

IN THE UNITED STATES DISTRICT COURT

**FOR THE NORTHERN DISTRICT
OF CALIFORNIA**

**ANIBAL RODRIGUEZ, SAL CATALDO,
JULIAN SANTIAGO, and SUSAN LYNN
HARVEY, individually and on behalf of
all other similarly situated,**

Plaintiffs,

V.

GOOGLE LLC,

Defendant.

§ 87(2)(b)

Case No. 3:20-cv-04688-RS

EXPERT REPORT OF JONATHAN E. HOCHMAN

March 22, 2023

Appendix I

WAA Test Apps

Appendix I – WAA Test Apps

Table of Contents

I. Configuring the Test Apps.....	2
II. Test Apps Operation	5
III. Unique Identifiers	7
IV. Data Collected by Firebase	9
V. Google’s BigQuery Data Warehouse.....	25
VI. Firebase Cloud Messaging.....	25

1. This Appendix describes an experiment process I undertook to better understand Google's collection of WAA-off data and sWAA-off data by way of Firebase. The experiment revolved around test apps that use Firebase, which my consultants developed, configured, and operated under my direction. The results of this process provide further support for the opinions I express in the main report about how Google, through Firebase, collects information from WAA-off and sWAA-off users' activity on non-Google apps, and how Google associates that data with unique identifiers. The process also revealed that Google provides no option for app developers to prevent Google from collecting (s)WAA-off data (short of disabling GA4F altogether), and moreover, that Google collects and saves data by way of GA4F that Google does not even share with app developers.

I. CONFIGURING THE TEST APPS

2. I directed the consulting experts listed in my main report to develop test apps for Android and iOS environments. Two sets of Test apps were developed, one set for Android and one set for iOS (collectively, "Test Apps").

3. A Google account, `rodriguez.experiment@gmail.com` was created on the `https://accounts.google.com` site. This account was used to create the Firebase account, Google Analytics account, Google Cloud account, and Apple ID/iCloud account.

4. The Test Apps incorporated the following Firebase features:

- Firebase Authentication
- Firebase Cloud Firestore
- Firebase Cloud Storage
- Google Analytics for Firebase
- Firebase Cloud Messaging
- Firebase Hosting

5. As part of this process, two Firebase projects: (1) "WAA - Toggle Experiment-0" (WAA0) and (2) "WAA - Toggle Experiment-1" (WAA1) were created. Similarly, two additional projects

were created for the Apple iOS WAA0 and WAA1 apps. This appendix includes exemplary screenshots, which are from the Android WAA0 app (unless otherwise stated).

6. Each project was configured to subscribe to Google Analytics for Firebase, as shown in Exhibit I-14.

7. Advertisements were incorporated in the Apps using an AdMob account. The ads were released ONLY in the debug mode.

8. Two different versions (WAA0 and WAA1) of the Test Apps were developed for each of the Android and iOS platforms.

9. The Test Apps allowed users to perform various activities such as:

- Log-in using Yahoo credentials or Test Apps own log-in credentials
- Perform simple eCommerce features, including:
 - browse items available for purchase
 - select an item to view details
 - add item to the cart
 - view the cart item(s)
 - remove an item from the cart
 - purchase item(s) from the cart
 - Receive cart action related notifications (via Cloud Messaging)
- The Test Apps included a button called “WebView Form” that linked to a screen that allowed the user to fill out a text field and submit the content. This function was developed using the WebView option of Firebase.
- Two buttons were added to the Test Apps with links to webpages to i) to “Browse Video” and ii) “Browse Form”.
 - The “Browse Video” screen webpage displayed a short video using youtube.com.
 - The “Browse Form” screen webpage allowed a user to enter information into a form. The form data collected was stored in the Firebase datastore.

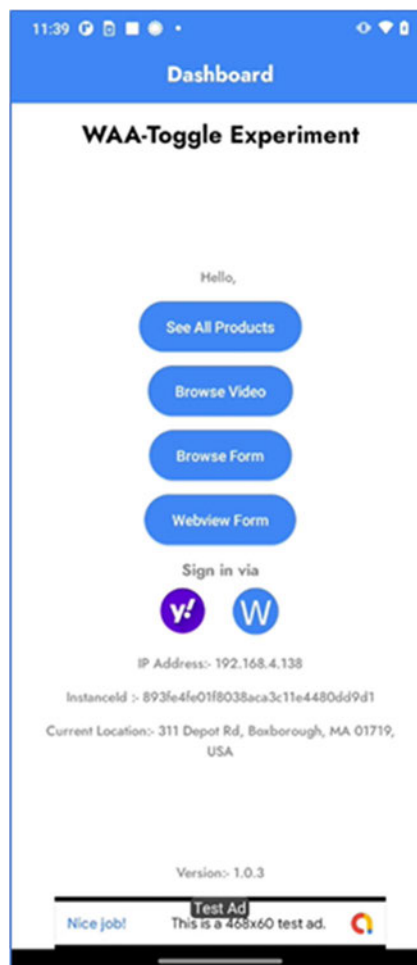
10. The Test Apps activated “Google Signals.”¹

¹ See [GA4] Activate Google Signals for Google Analytics 4 properties, Analytics Help, <https://support.google.com/analytics/answer/9445345?hl=en#zippy=%2Cin-this-article> (Last accessed March 21,

11. The following screenshots are the home screens for the WAA0 and WAA1 apps.



WAA0 Android App



WAA1 Android App

Additional exemplary screenshots and video of the Android WAA0 App are included in Exhibit I-12.

12. The APK (Android Package Kit) of the current versions of the Android test App APKs is included in Exhibit I-3.

2023); *Activate Google Signals*, Analytics Help (google.com), <https://support.google.com/analytics/answer/7532985?hl=en#zippy=%2Cin-this-article> (Last accessed March 21, 2023).

13. The Android Test Apps were installed on a “Moto G Play” phone running Android V11, IMEI: 355567114386039, Google Advertisement ID: 01ba3d4c-e5c6-49fc-88f7-c34286f9600e. The phone was configured using the test1.rodriquez@gmail.com account.
14. The Apple iOS WAA Apps were deployed on an iPad model running iOS V16.3.1 device. The device was configured using test3.rodriquez@gmail.com as the Apple ID. The WAA Apps were installed using TestFlight App.²
15. Exemplary screenshots of how the Apple iOS WAA0 App was operated are included in Exhibit I-12.
16. The IPA of the current versions of Apple iOS Test Apps are included in the Exhibit I-13.

II. TEST APPS OPERATION

17. The Test Apps were operated in sessions as described below, on both Android and iOS.:
- Launch WAA0 App
 1. Sign-in using Yahoo account and perform following steps:
 - Browse eCommerce item
 - Add an item to the cart
 - Remove the item from the cart
 - Purchase another item
 - Watch a video using the web link provided
 - Complete the form using the web link provided and submit the data
 - Click on an ad shown on the main screen
 2. Sign out of the Yahoo account
 3. Sign-in using WAA-App specific account credentials
 4. Repeat the steps listed above for the Yahoo signed-in phase
 5. Sign out of the WAA-App specific account
 6. Repeat the steps listed above for the Yahoo signed-in phase while not signed into any account
 7. Close the WAA0 App
 - Launch WAA1 App and repeat the steps listed above for WAA0 App.

² See, *Beta Testing Made Simple with TestFlight*, App Store Connect – Apple Store Developer, <https://developer.apple.com/testflight/> (Last accessed March 21, 2023).

18. Exemplary screenshots of how the Android WAA0 App was operated using Yahoo account sign-in credentials are included in Exhibit I-11.

19. The Moto G Play phone was set to a USB debugging mode and connected to a laptop that had a “Debugging Proxy Tool” called “Fiddler Classic” installed and running (<https://www.telerik.com/download/fiddler>).

20. The steps listed for the WAA0 app in paragraph ¶ 17 were performed while the Fiddler Classic tool was running, and its traffic files were saved.

21. The Fiddler Classic captures file is included in Exhibit I-10.

22. Table IA below lists an exemplary schedule and status of the WAA and sWAA settings.

The screenshots showing these changes are available in Exhibit I-9.

Date	Time (ET)	WAA	SWAA
3/14/2023	05.24	ON	ON
3/15/2023	13.20	ON	OFF
3/16/2023	12.06	OFF	OFF

Table IA

23. The WAA Apps were operated as described in paragraph ¶ 17 after the WAA and sWAA settings were toggled. Table IB below lists the schedule when the Moto G Play phone using the test1.rodriguez@gmail.com account was operated.

Date	Time (ET)	APK ID	WAA	SWAA
03/15/2023	~11:05	WAA0	ON	ON
03/15/2023	~11:20	WAA1	ON	ON
03/15/2023	~18.25	WAA0	ON	OFF
03/15/2023	~18.40	WAA1	ON	OFF
03/16/2023	~18.15	WAA0	OFF	OFF
03/16/2023	~18.25	WAA1	OFF	OFF

Table IB

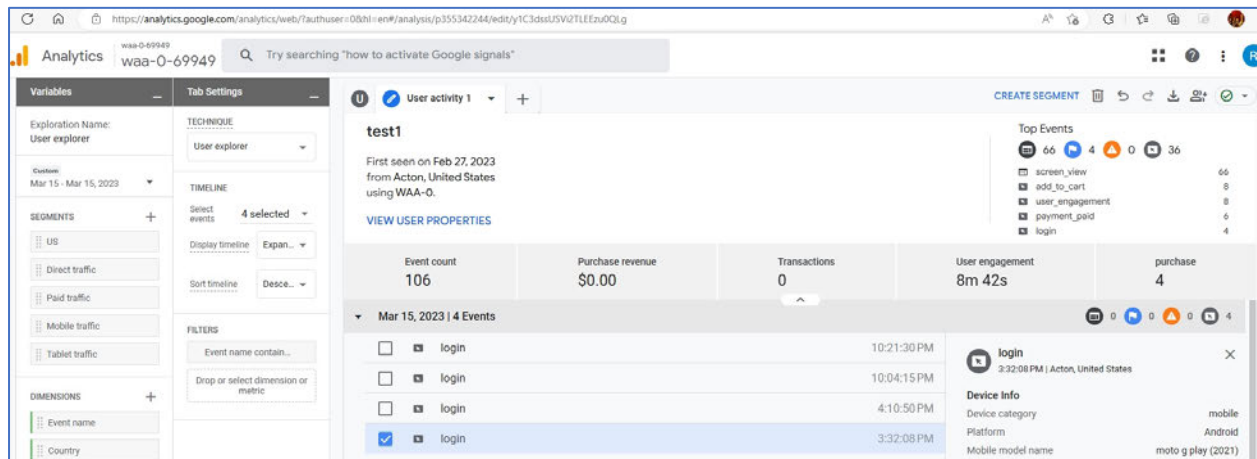
III. UNIQUE IDENTIFIERS

24. Firebase enables Google to assign unique identifiers to users tied to their signed-in identifies on apps, regardless of WAA and sWAA status.

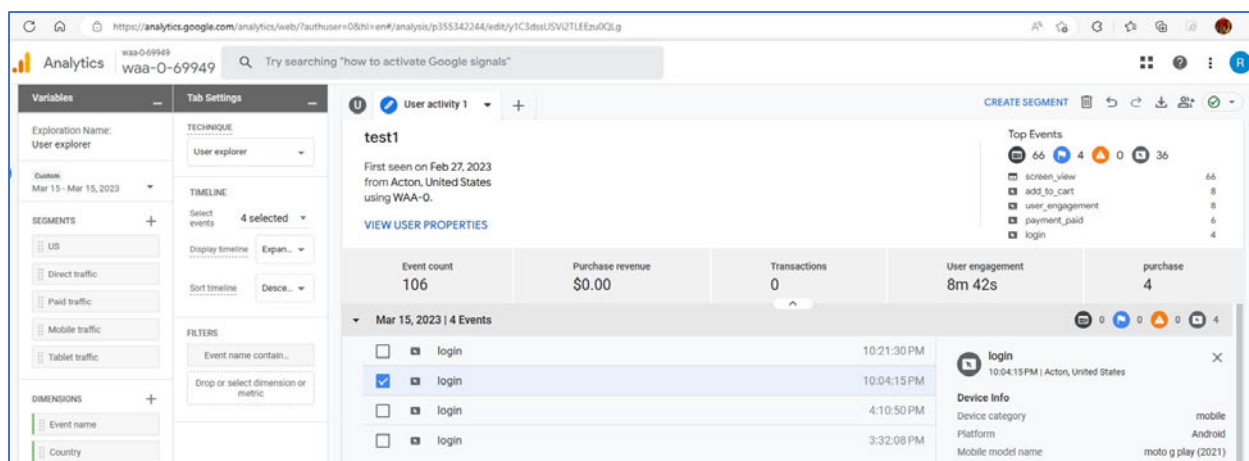
25. The Firebase SDK Authenticate feature was configured to allow users to log into the Apps using the Yahoo Federated Identity Provider (authentication.account@yahoo.com). When a user does so, the Firebase Authenticate assigns a specific “uid” value for the signed-in user.

26. The Firebase SDK Authenticate feature was also configured to allow users to log into the Apps with specific login credentials. When a user logs in with specific credentials (e.g., “test1”), the Firebase Authenticate assigns a different “uid” value (“test1”) for the signed-in user.

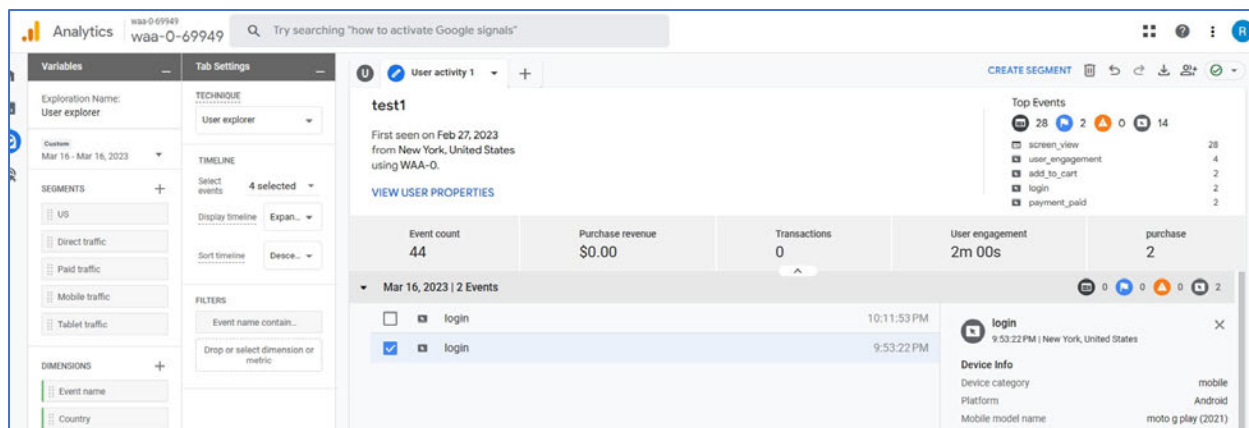
27. Here is a screenshot showing the value “test1” assigned as the “uid” for a logged-in user when both WAA and sWAA are turned on:



28. Here is a screenshot showing the value “test1” assigned as the “uid” for a logged-in user when WAA is on but sWAA is off:



29. Here is a screenshot showing the value “test1” assigned as the “uid” for a logged-in user when WAA and sWAA are turned off:



30. The “WAA0” test App for Android received the user ID (“uid”) value from the Firebase Authorization Admin SDK and used this returned “uid” to set the “user_id” value for the BigQuery database to associate analytics data with the same user by using `mFirebaseAnalytics.setUserId()` method³

31. The “WAA1” test App for Android did not invoke the `mFirebaseAnalytics.setUserId()` method to set the `userId`. Firebase assigned the app instance identifier as the user ID.

³ See *Set a user ID*, Google Analytics for Firebase – Documentation, <https://firebase.google.com/docs/analytics/userid#android> (Last accessed March 21, 2023).

IV. DATA COLLECTED BY FIREBASE

32. Firebase provides app developers with a console to configure apps. The console also contains a Firebase Dashboard that allows app developers to view, query, and manage various user data received by Google. The Firebase dashboard also includes the Firebase Analytics Dashboard.

33. A review of the Firebase dashboard confirms that Google servers receive and store data generated from users' activity on the app regardless of the user's WAA and sWAA status.

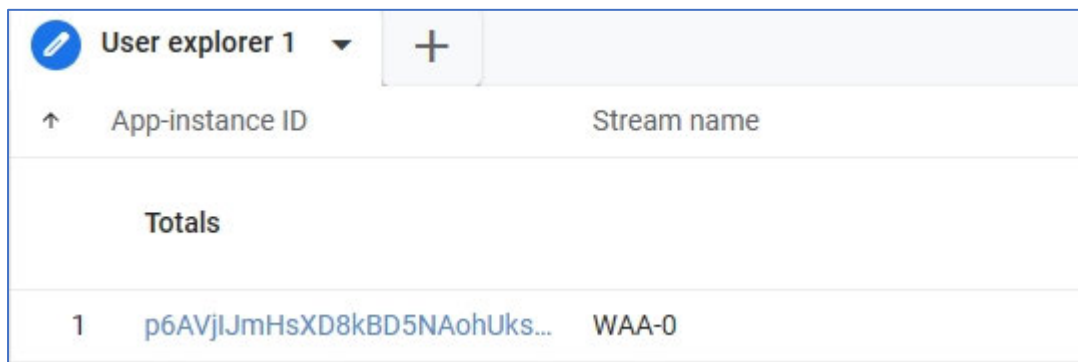
34. Moreover, a review of all the configuration options available on Firebase, including documentation made available by Google, did not reveal any API calls, functions, or source code snippets that would allow app developers to ascertain the WAA/sWAA status of users. Short of disabling GA4F collection altogether for all users, app developers have no way to prevent Google from collecting data from WAA-off and sWAA-off users. Firebase does not provide app developers with the ability to selectively enable or disable Firebase or Analytics collection (of events and data) based on the WAA or sWAA status of app users.

35. Firebase assigned a unique identifier that is used to associate the captured data as follows:

- When a user has not logged in to the WAA App, Firebase Analytics associates Analytics collection to the "app instance id" ("appInstanceID") assigned to the app on the device. This app instance identifier is a unique identifier assigned to each Android app installed on a specific device, regardless of WAA and sWAA status. Further experiments indicate that the app instance identifier remained the same, even when the app was updated to a new version of the same app on the same device (see Exhibit I-8), and independent of WAA or sWAA status.
- When a user logs in using log-in credentials (e.g., Yahoo or WAA-Apps specific login credentials), Firebase Authenticate assigned a unique identifier value as the "uid" based on the credentials as described above.

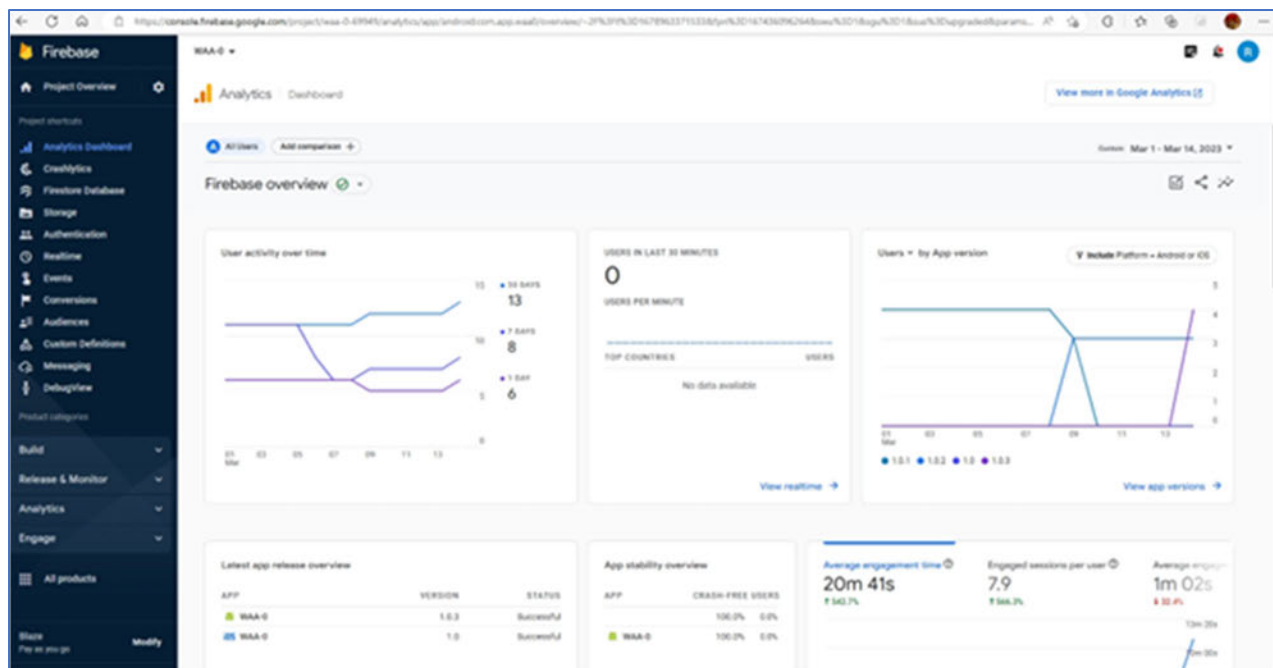
- The same unique identifier was consistently assigned by Firebase Authenticate across multiple app sessions.

36. Additionally, the same unique “uid” value was assigned by Firebase to a user when the user logged in on different devices using the same login credentials. For example, the authentication.account@yahoo.com account was used to log in to the WAA0 App on at least three different devices (Moto G Play phone, Realme Pad tablet running Android operating system, and Apple iPad running iOS operating system) and in each instance, the same “uid” value was assigned to the user. See the screenshot below.



	App-instance ID	Stream name
Totals		
1	p6AVjIJmHsXD8kBD5NAohUks...	WAA-0

37. An analysis of the Firebase dashboard also indicates that Firebase collects data related to the app and web related activities of the user using the unique identifier. See exemplary screenshots below.



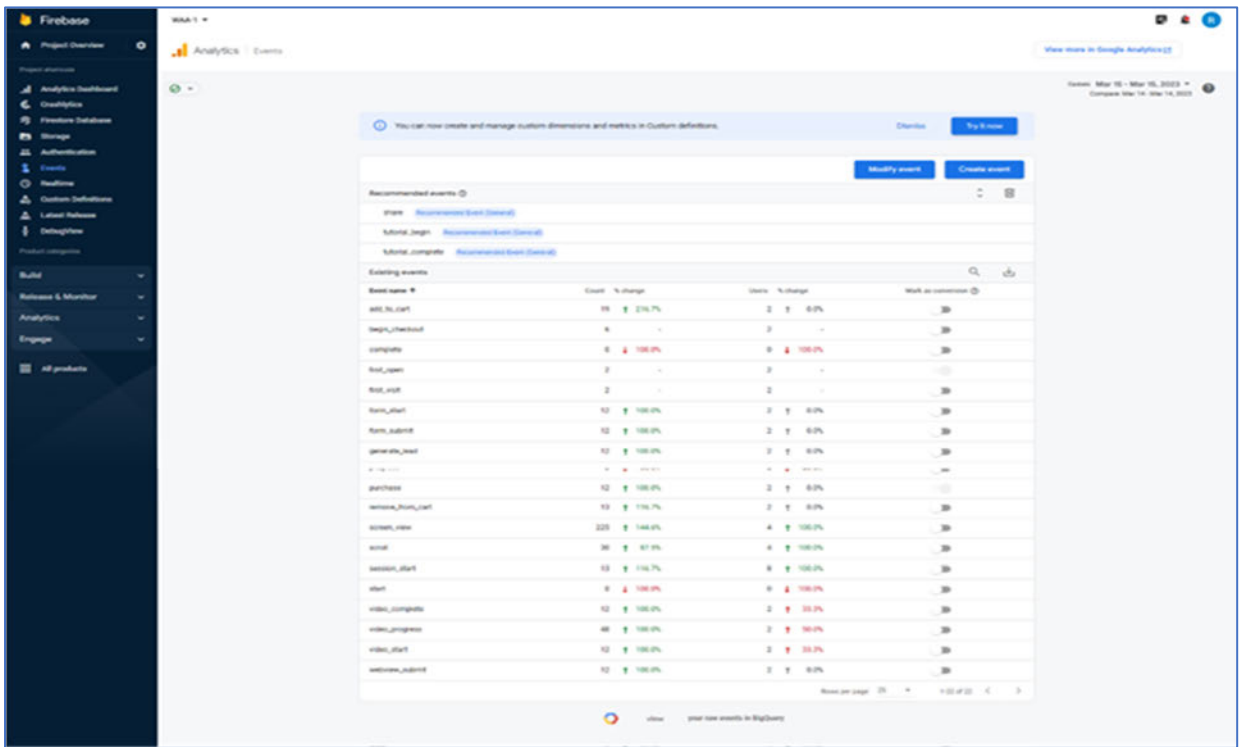
The screenshot shows the Firebase Analytics dashboard for app WAA-0, specifically the 'Events' section. The dashboard includes a sidebar with navigation options like Project Overview, Analytics Dashboard, Crashlytics, and more. The main content area displays a table of existing events:

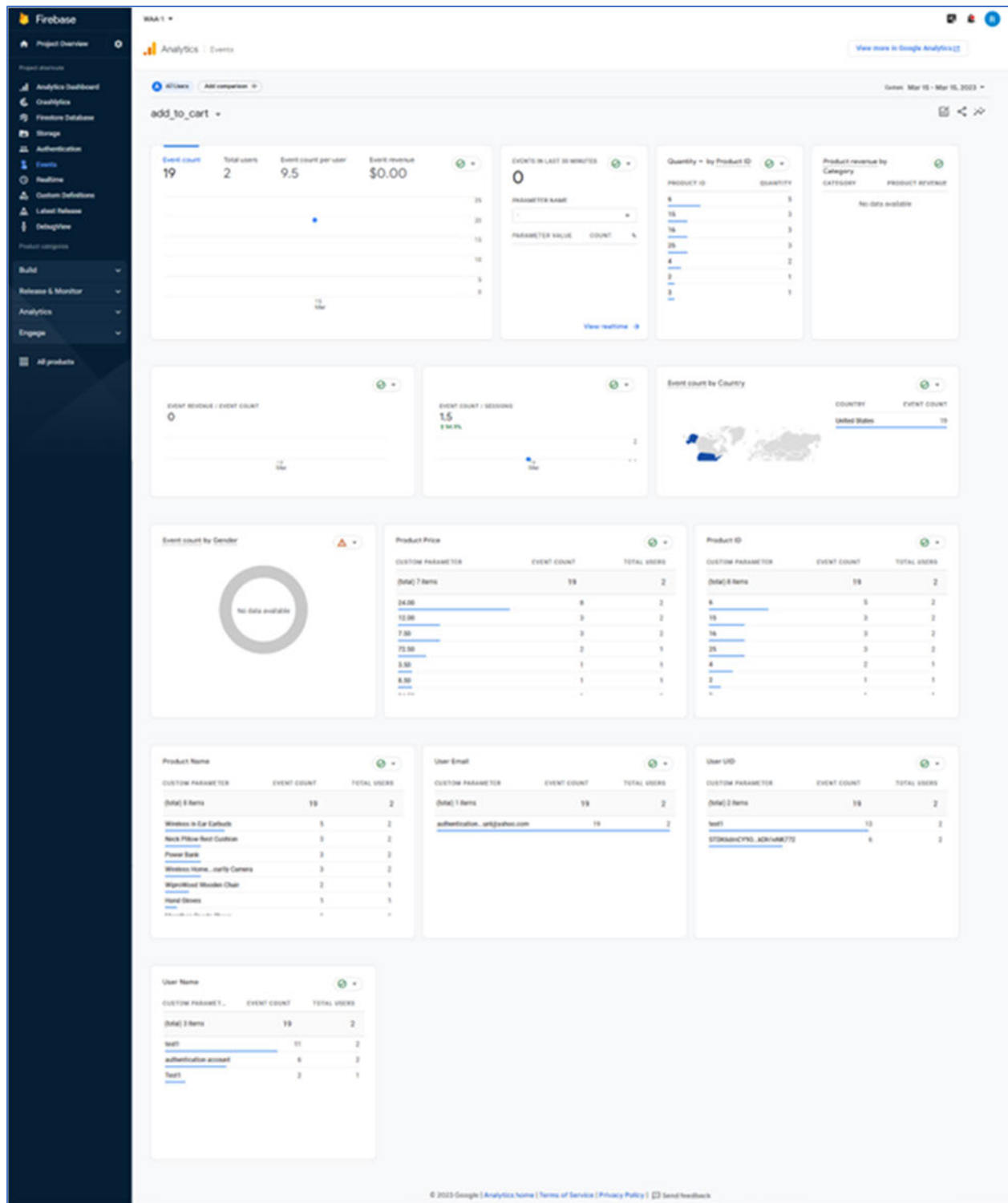
Event name	Count	% change	Users	% change	Mark as conversion
add_to_cart	64	↑ 481.0%	6	↑ 50.0%	
app_remove	1	↓ 66.7%	1	↓ 66.7%	
app_update	0	↓ 100.0%	0	↓ 100.0%	
complete	64	↑ 481.0%	5	↑ 25.0%	
first_login	0	↓ 100.0%	0	↓ 100.0%	
first_visit	0	↓ 100.0%	0	↓ 100.0%	
form_start	63	↑ 567.0%	2	↑ 50.0%	
form_submit	63	↑ 567.0%	2	↑ 50.0%	
generate_lead	63	↑ 567.0%	2	↑ 50.0%	

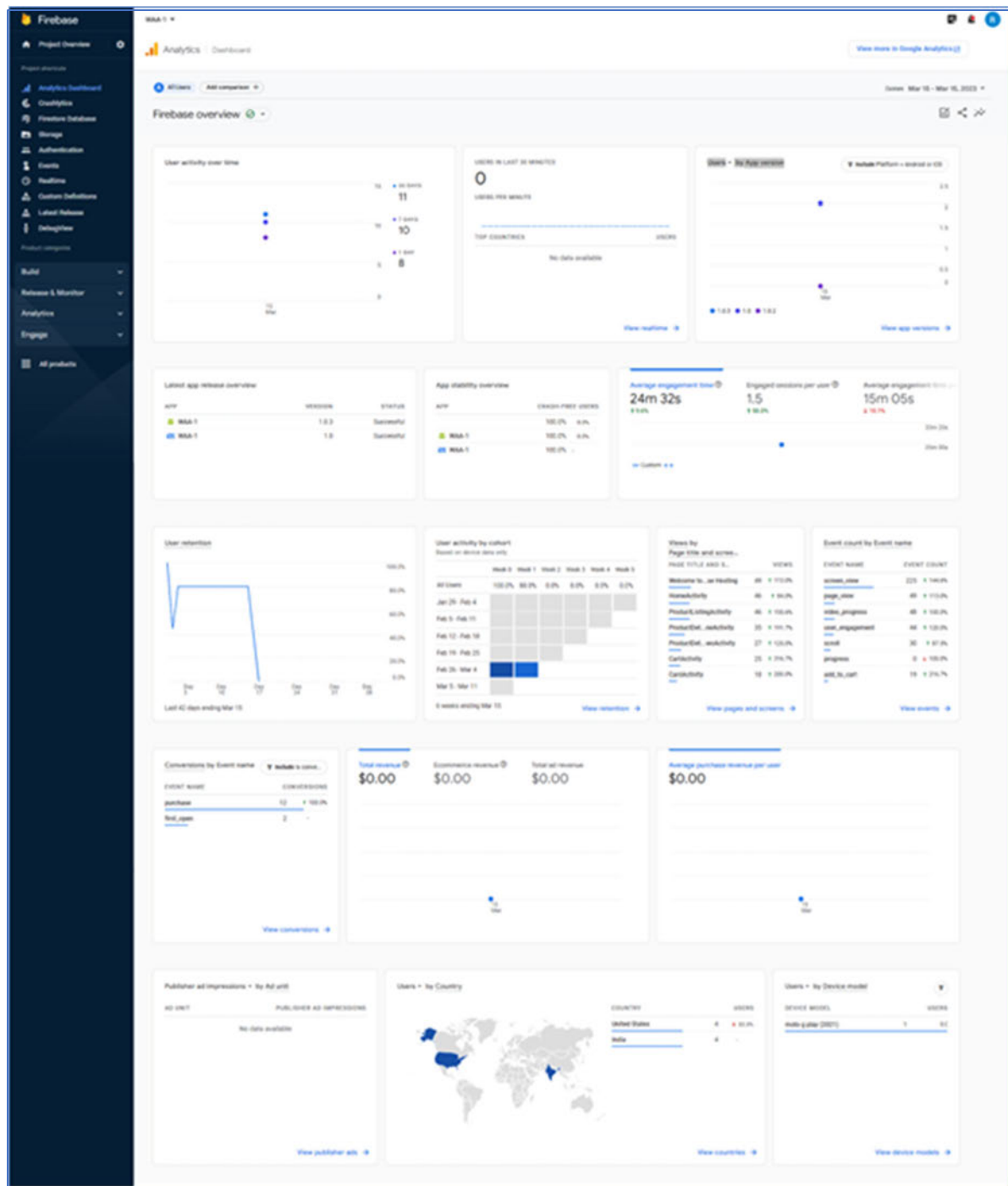
38. Exhibit I-1 lists events and parameters that were automatically collected by Firebase.
39. Exhibit I-2 lists exemplary programmed events for the WAA0 and WAA1 Apps.
40. App activity and events data were captured by Firebase across the days of the experiment process under different WAA and sWAA setting states and combinations, as I now summarize.

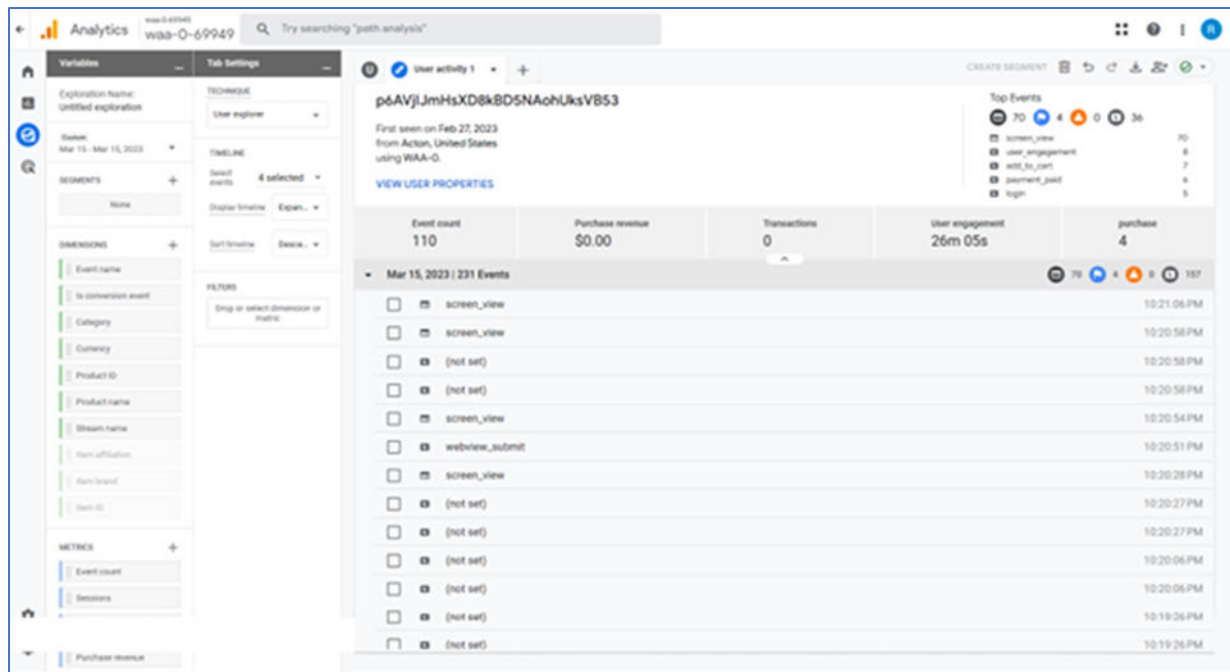
WAA “On” and sWAA “On”

41. The Moto G Play phone was operated on March 15, 2023 at around 2:00 PM GMT, with WAA turned on and sWAA turned on. The following exemplary screenshots show the events being captured by Firebase:

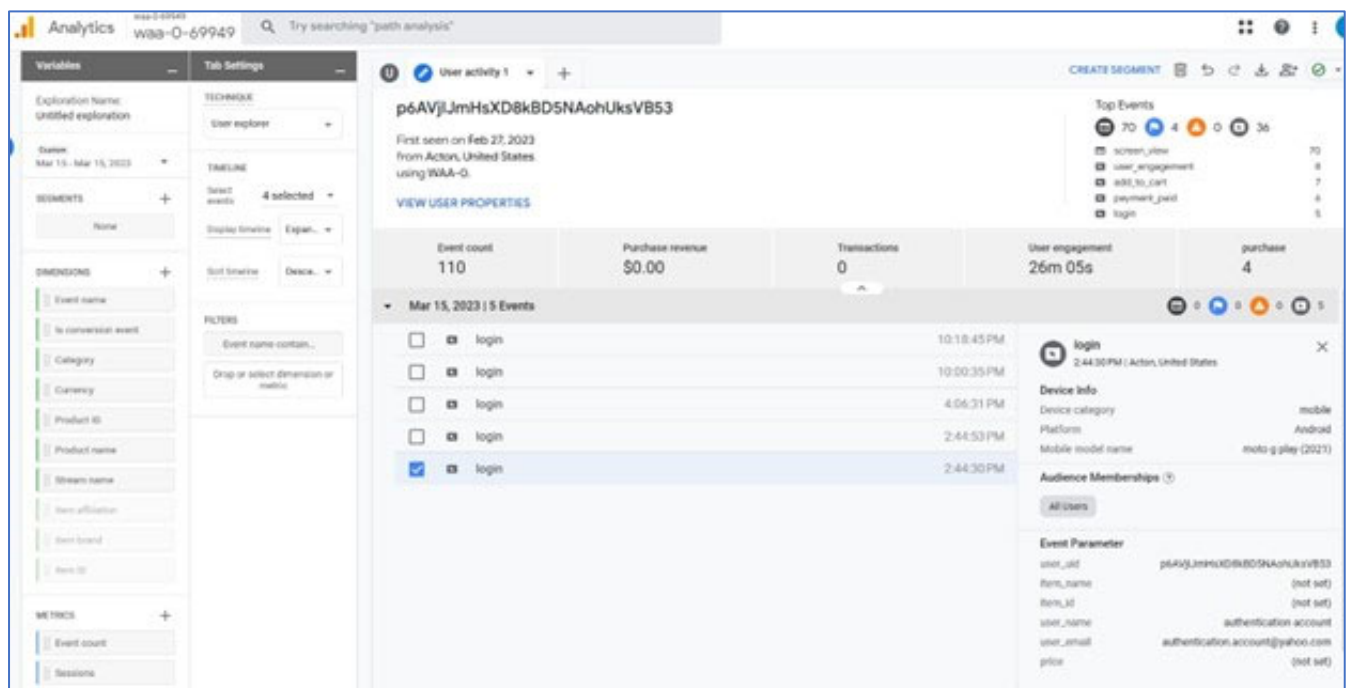




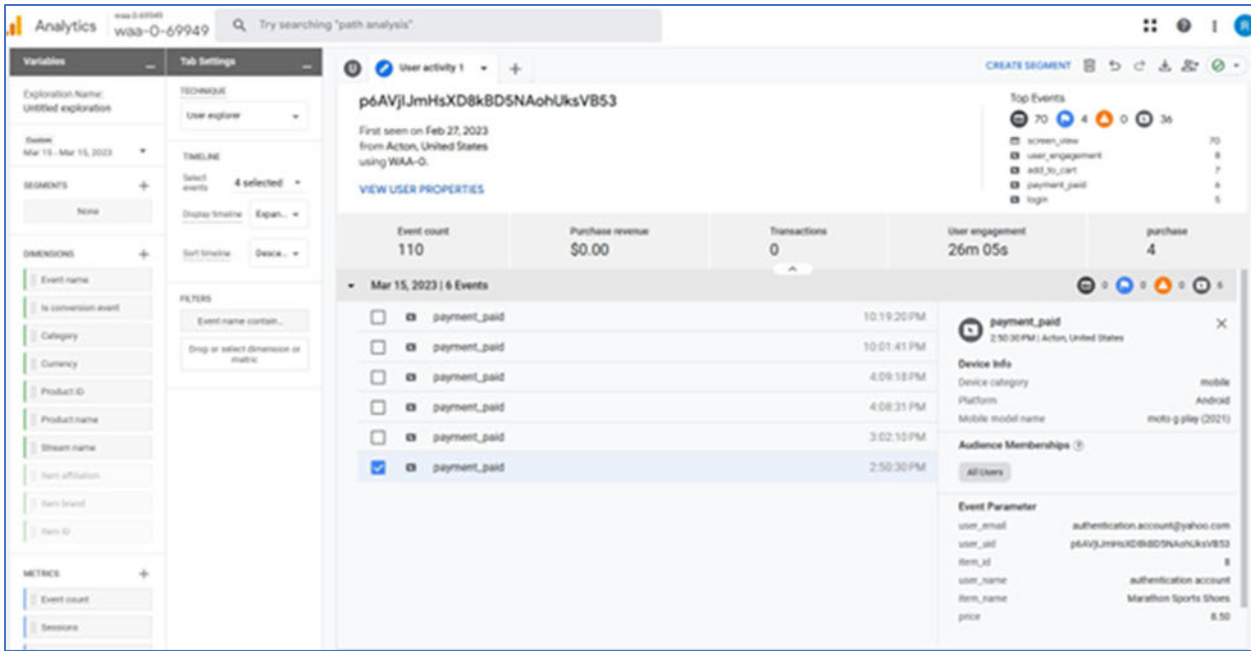
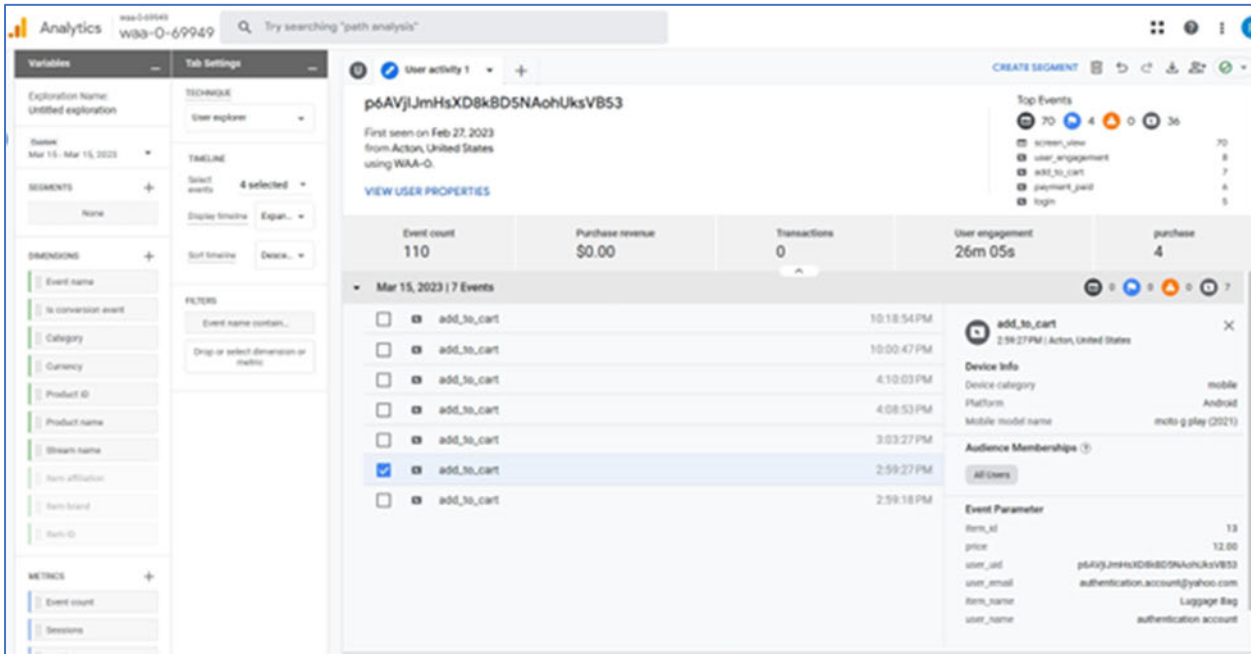




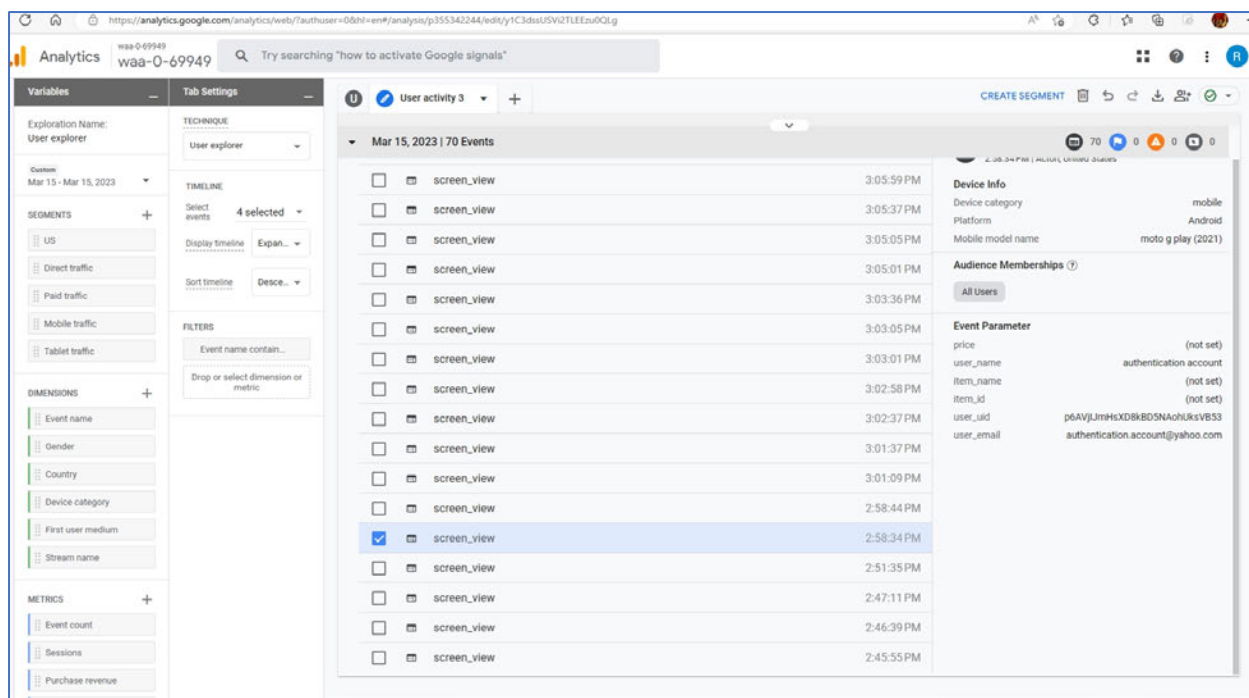
42. The following login event was automatically transmitted by Firebase for the user of the WAA0 App while logged in to the app using the authentication.account@yahoo.com account:



43. Once logged in, the user’s add_to_cart and payment_paid events were collected by Google including with the following information: device category, platform, mobile device name, product details, user id, and user email address.



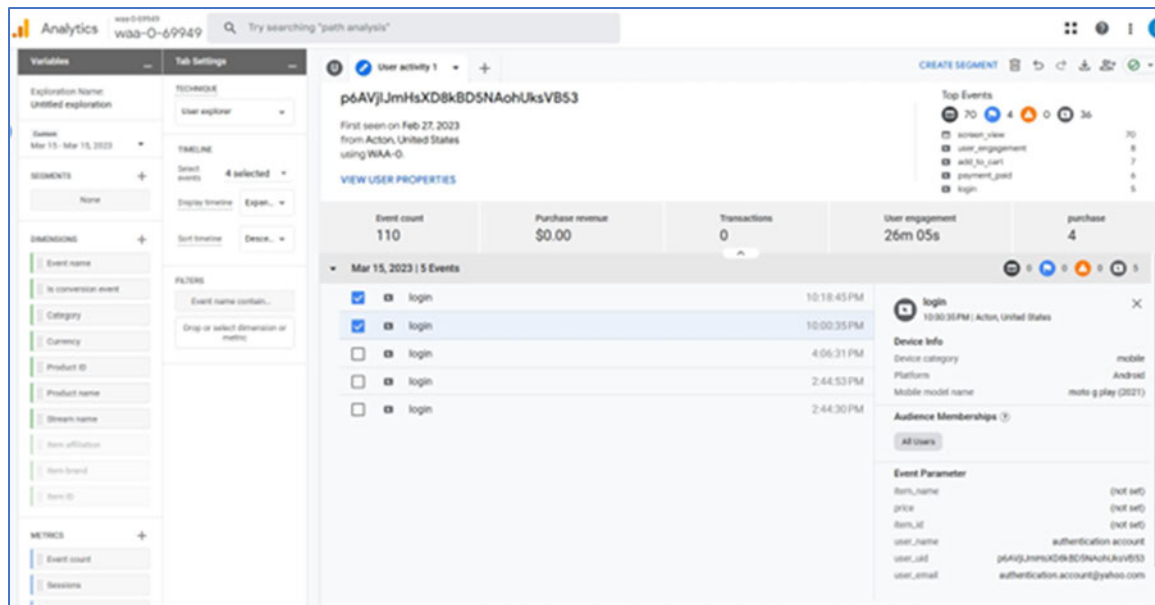
44. The following screenshot shows the collection of various events transmitted by Firebase automatically.



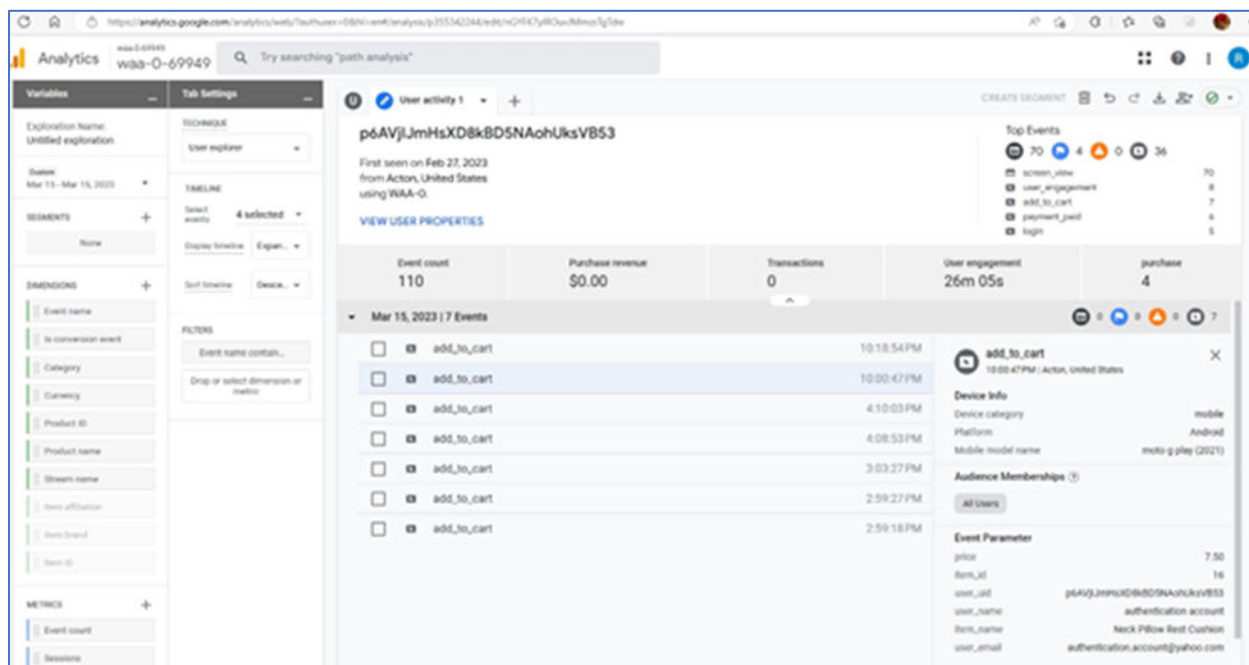
45. Additional screenshots and video of the additional reports available on the Firebase dashboard for such activities of the Moto G phone operated on March 15, 2023, with WAA on and sWAA on are included in Exhibit I-4.

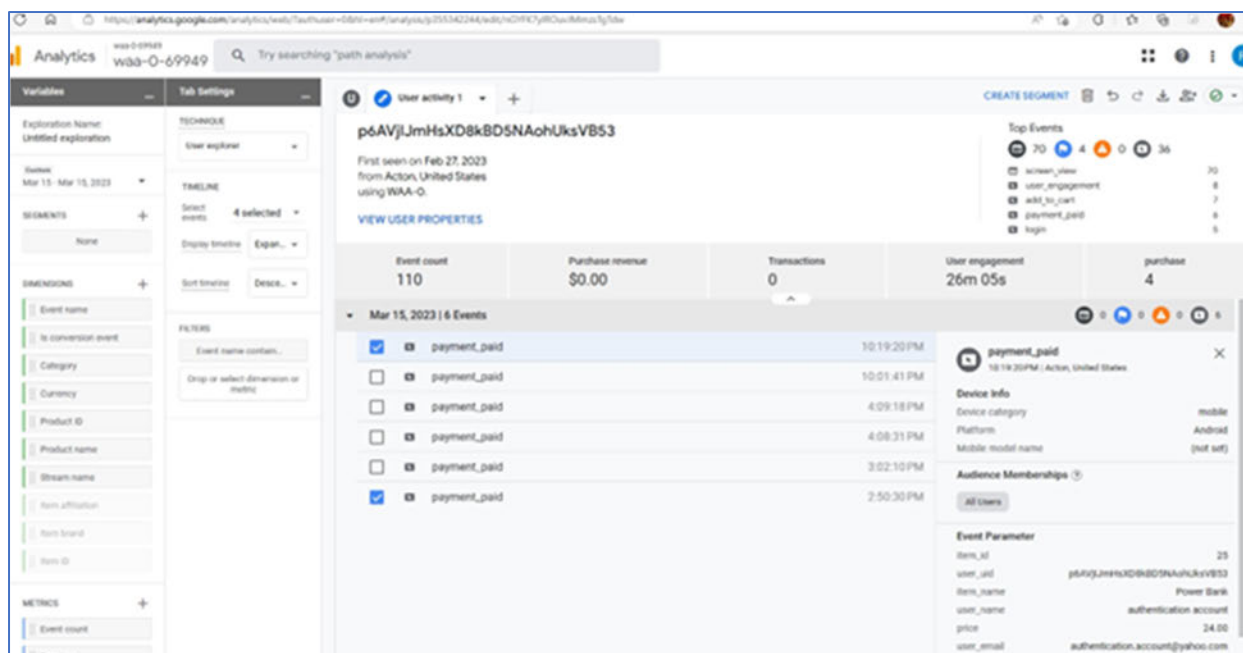
WAA “On” and sWAA “Off”

46. When the Moto G phone was operated on March 15, 2023 at around 10:20 PM GMT, with the WAA on and sWAA off, the following login event was automatically transmitted by Firebase for the user of the WAA0 App logged in using the authentication.account@yahoo.com account, as shown below:

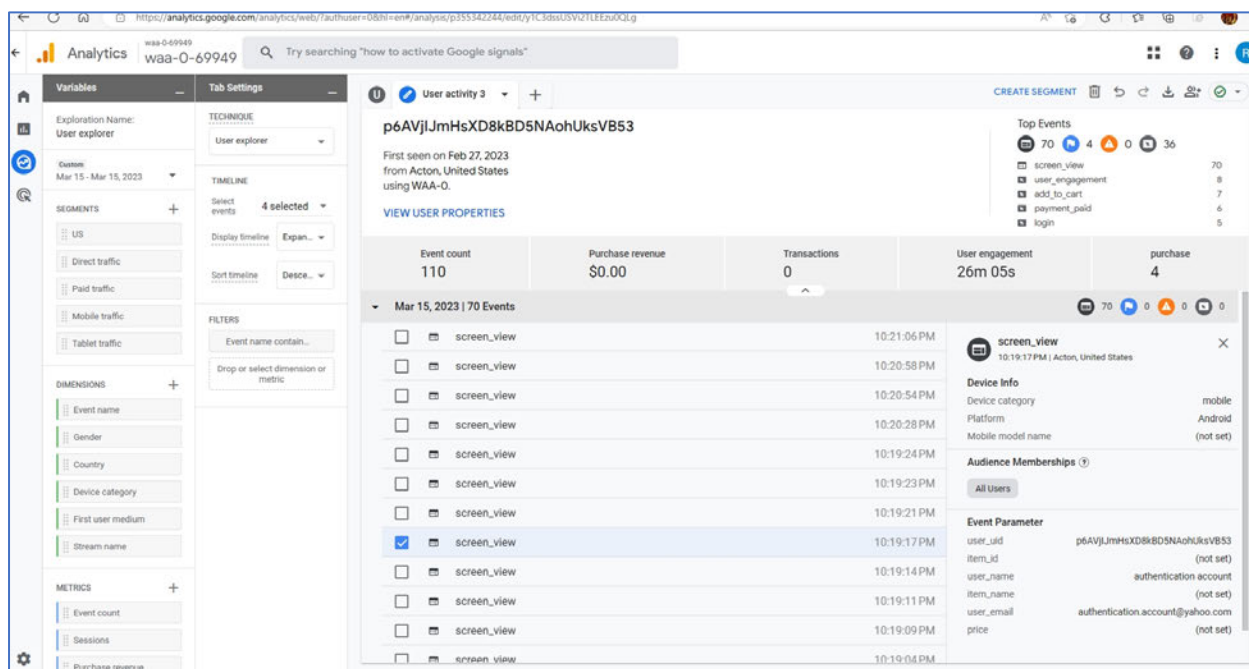


47. Once logged in, the user's add_to_cart and payment_paid events were collected by Google, along with the following information: device category, platform, mobile device name, product details, user id, and user email address—as shown in the following screenshots:





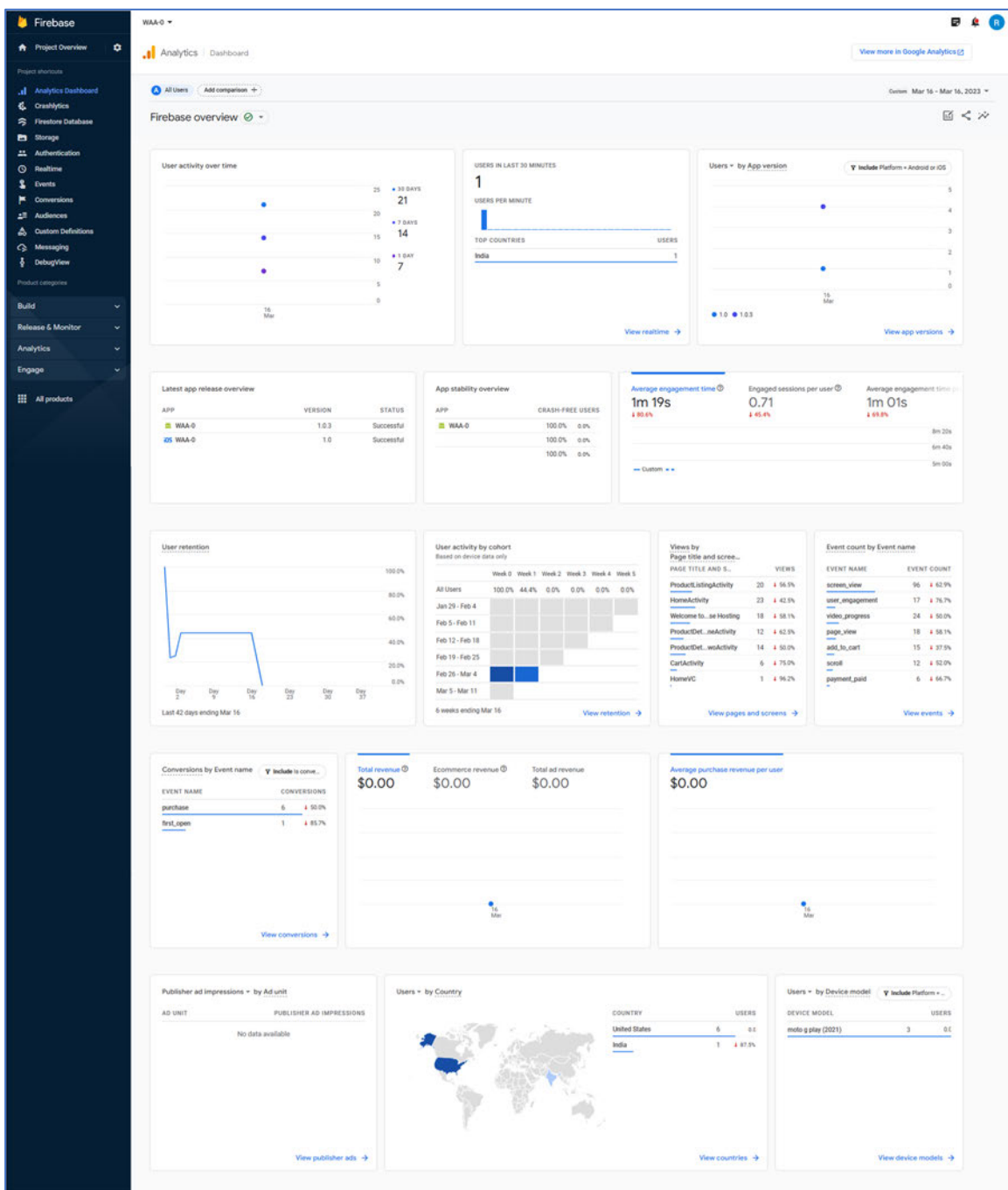
48. The following screenshot shows the collection of various screen_view events transmitted by Firebase automatically.



49. Additional screenshots and video of the reports available on the Firebase dashboard for such activities of the Moto G phone, operated on March 15, 2023 at around 10:0 PM GMT, with the WAA on and sWAA off are included in Exhibit I-5.

WAA “Off” and sWAA “Off”

50. The Moto G phone was operated on March 16, 2023 at around 9:40 PM GMT, with WAA off and sWAA on. The following screenshots depict data captured by Firebase:



51. The following login event was automatically collected by Firebase for the WAA0 App user logged in using the authentication.account@yahoo.com account, as shown below:

Analytics waa-0-69949 Try searching "how to activate Google signals"

Variables **Tab Settings**

Exploration Name: User explorer

Custom: Mar 16 - Mar 16, 2023

SEGMENTS +

- US
- Direct traffic
- Paid traffic
- Mobile traffic
- Tablet traffic

DIMENSIONS +

- Event name
- Gender
- Country
- Device category
- First user medium
- Stream name

METRICS +

- Event count
- Sessions

TECHNIQUE User explorer

TIMELINE Select events: 4 selected Display timeline: Expan... Sort timeline: Desc...

FILTERS Event name contain... Drop or select dimension or metric

User activity 3 +

p6AVjJmHsXD8kBD5NAohUksVB53

First seen on Feb 27, 2023 from New York, United States using WAA-0.

VIEW USER PROPERTIES

Event count	Purchase revenue	Transactions	User engagement	purchase
59	\$0.00	0	2m 43s	2

Mar 16, 2023 | 2 Events

Event	Time
login	10:08:59 PM
login	9:49:23 PM

login 9:49:23 PM | New York, United States

Device Info

- Device category: mobile
- Platform: Android
- Mobile model name: moto g play (2021)

Audience Memberships All Users

Event Parameter

- user_id: p6AVjJmHsXD8kBD5NAohUksVB53
- item_id: (not set)
- user_email: authentication.account@yahoo.com
- item_name: (not set)
- user_name: authentication account
- price: (not set)

Top Events

- screen_view: 33
- add_to_cart: 11
- user_engagement: 5
- login: 2
- payment_paid: 2

52. Once logged in, the user's add_to_cart and payment_paid events were collected by Google, along with the following information: device category, platform, mobile device name, product details, user id, and user email address..

Analytics waa-0-69949 Try searching "how to activate Google signals"

Variables **Tab Settings**

Exploration Name: User explorer

Custom: Mar 16 - Mar 16, 2023

SEGMENTS +

- US
- Direct traffic
- Paid traffic
- Mobile traffic
- Tablet traffic

DIMENSIONS +

- Event name
- Gender
- Country
- Device category
- First user medium
- Stream name

METRICS +

- Event count
- Sessions

TECHNIQUE User explorer

TIMELINE Select events: 4 selected Display timeline: Expan... Sort timeline: Desc...

FILTERS Event name contain... Drop or select dimension or metric

User activity 3 +

p6AVjJmHsXD8kBD5NAohUksVB53

First seen on Feb 27, 2023 from New York, United States using WAA-0.

VIEW USER PROPERTIES

Event count	Purchase revenue	Transactions	User engagement	purchase
59	\$0.00	0	2m 43s	2

Mar 16, 2023 | 11 Events

Event	Time
add_to_cart	10:09:13 PM
add_to_cart	9:50:40 PM
add_to_cart	9:50:05 PM
add_to_cart	9:50:05 PM
add_to_cart	9:50:04 PM
add_to_cart	9:50:03 PM
add_to_cart	9:50:00 PM
add_to_cart	9:50:00 PM
add_to_cart	9:49:59 PM
add_to_cart	9:49:53 PM
add_to_cart	9:49:33 PM

add_to_cart 9:50:03 PM | New York, United States

Device Info

- Device category: mobile
- Platform: Android
- Mobile model name: moto g play (2021)

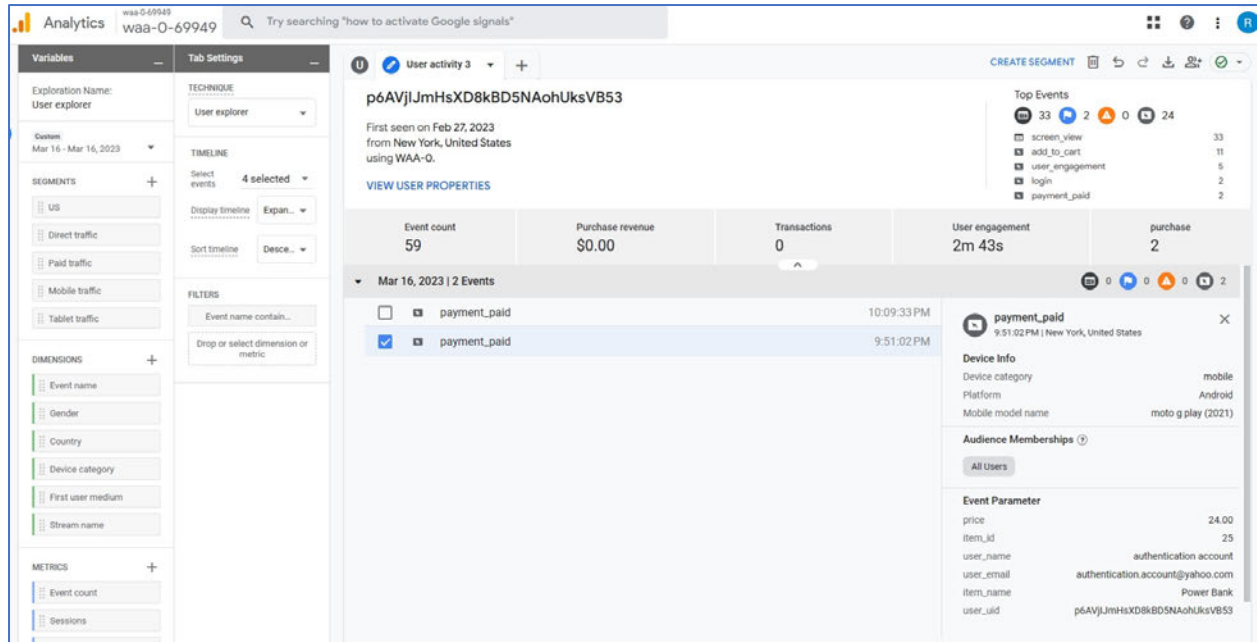
Audience Memberships All Users

Event Parameter

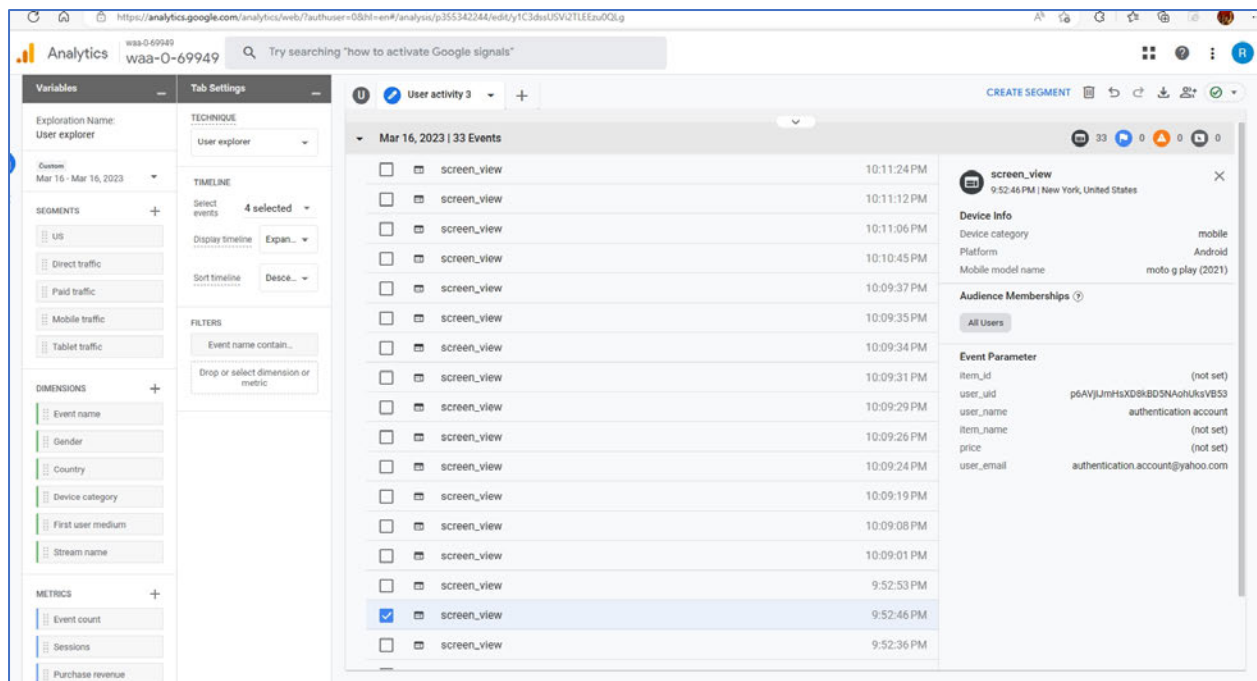
- item_id: 16
- user_name: authentication account
- price: 7.50
- item_name: Neck Pillow Rest Cushion
- user_id: p6AVjJmHsXD8kBD5NAohUksVB53
- user_email: authentication.account@yahoo.com

Top Events

- screen_view: 33
- add_to_cart: 11
- user_engagement: 5
- login: 2
- payment_paid: 2

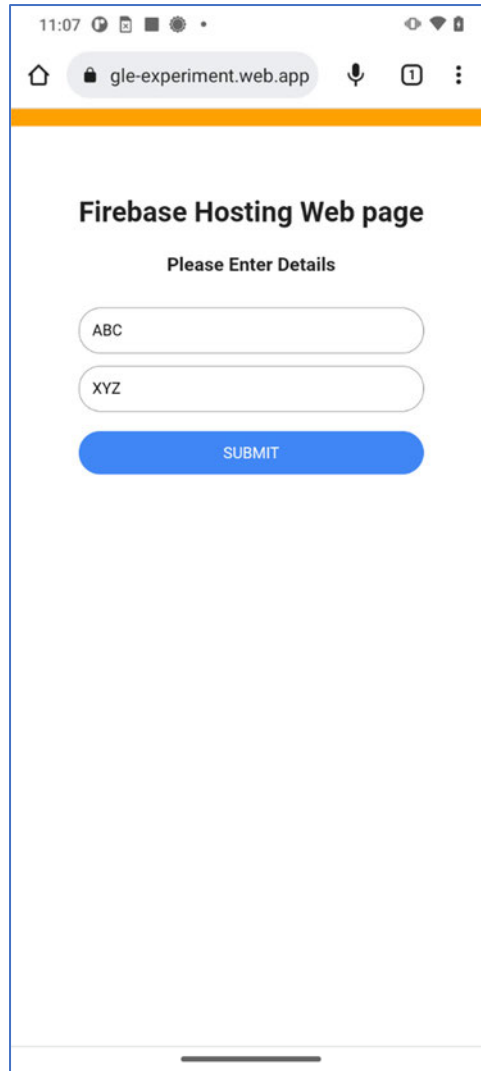


53. The following screenshot shows the collection of various screen_view events collected by Firebase automatically.



54. Additional screenshots and video of the reports available on the Firebase dashboard for such activities of the Moto G phone, operated on March 16, 2023 at around 9:40 PM GMT, with the WAA off and sWAA off are included in Exhibit I-6.

55. I also studied the data collected by Google when a user visited a web page linked within the app. Google, through Firebase, captured the full URL of the page. For example, when the user clicks on the “Browse Form” button, the App allows the user to fill out the form located at the “https://waa-toggle-experiment.web.app/ga-form?id=YzU5NDViZjQxYmI5NzE3MzRkOGJmMDgzZWYwNDZhNDc=&key=waa0” URL. See the screenshot below:



56. Firebase automatically generates the “page_view” event and collects the session id, page title, and the whole URL of the page visited, and automatically transmits the Page_view” event to Google.

* * *

57. I repeated these experiments to analyze the events captured by Firebase for both logged-in and logged-out sessions (meaning logged in on the app). I also repeated these experiments for the iOS Test Apps. My conclusions are:

- i) Firebase continues to collect app activity even when WAA and/or sWAA are off. Switching those settings “on” and “off” has no material effect on the data collected by Firebase.
- ii) Whether WAA and sWAA are on or off, Firebase continues to collect app activity regardless of whether the user signs into the app. The only difference is that, when a user signs in, Firebase associates the data with a unique app-focused identifier.
- iii) Firebase collected the same data from users’ app activity on the Android Test Apps as the iOS Test Apps.

V. CLOUD MESSAGING

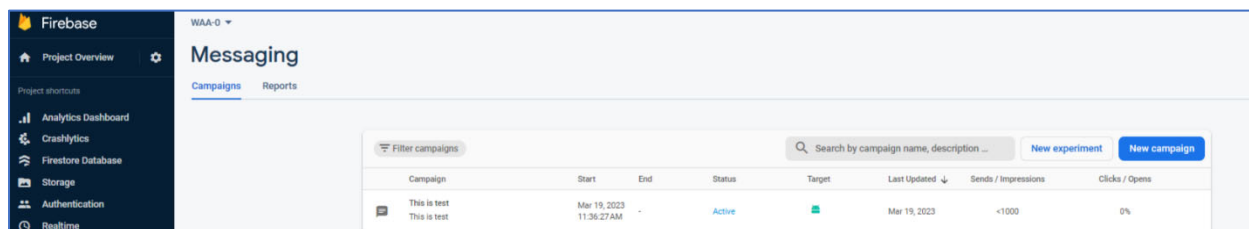
58. This testing process has also shown that Google collects data through Firebase Cloud Messaging (FCM) regardless of WAA and sWAA status. FCM automatically collects the following information.⁴

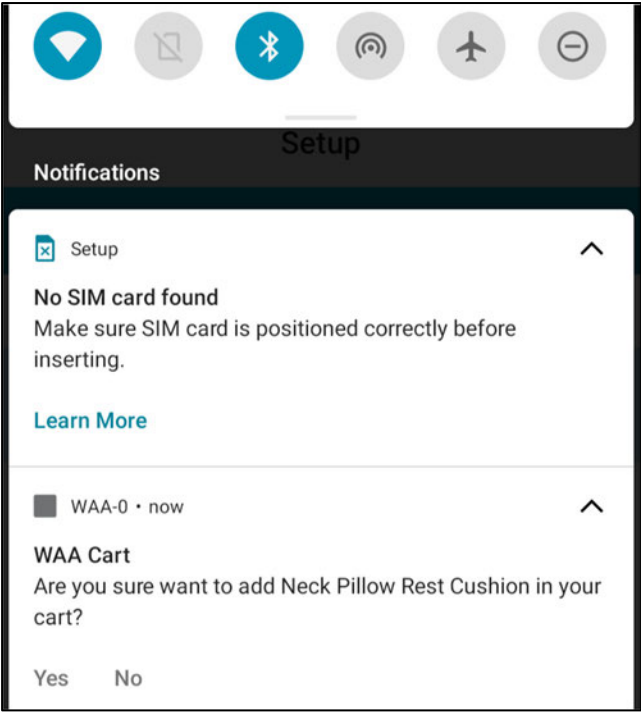
⁴ See [GA4] *Automatically Collected Events*, Firebase Help (google.com), <https://support.google.com/firebase/answer/9234069> (Last accessed March 21, 2023)

notification_dismiss (app)	when a user dismisses a notification sent by Firebase Cloud Messaging (FCM) Android only	message_name, message_time, message_device_time, message_id, topic, label, message_channel
notification_foreground (app)	when a notification sent by FCM is received while the app is in the foreground	message_name, message_time, message_device_time, message_id, topic, label, message_channel, message_type
notification_open (app)	when a user opens a notification sent by FCM	message_name, message_time, message_device_time, message_id, topic, label, message_channel
notification_receive (app)	when a notification sent by FCM is received by a device when the app is in the background Android only	message_name, message_time, message_device_time, message_id, topic, label, message_channel, message_type

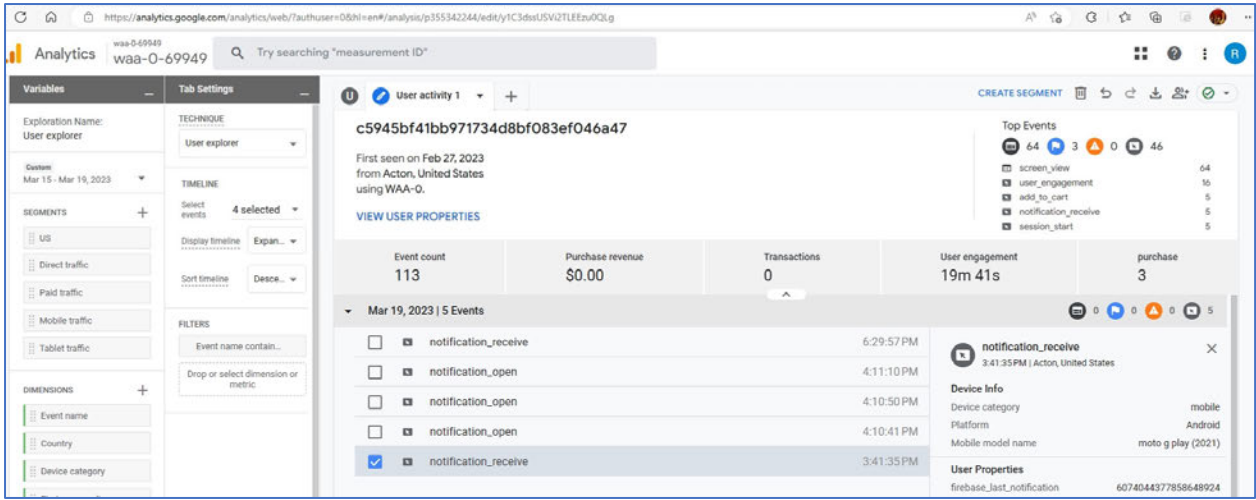
59. Experiments with the WAA0 Apps show that Google collects the notification-related data and events regardless of the WAA/sWAA status of the user.

60. For example, a campaign was created using the Firebase console and notifications were sent to the WAA0 App.

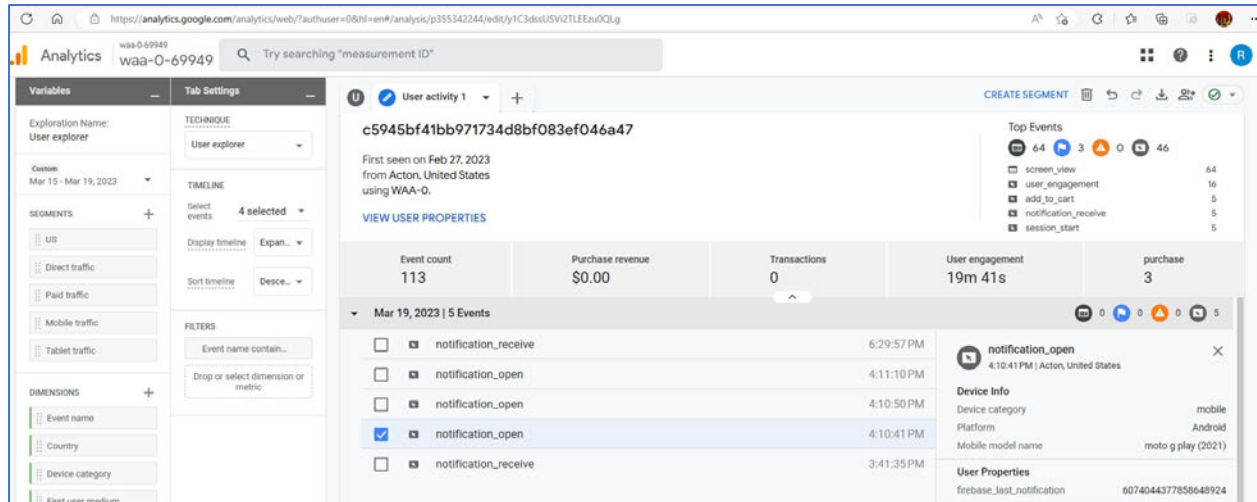




61. When the notification was received by the Moto G phone with WAA0 App installed, Google collected the “notification_receive” event regardless of WAA and sWAA state.



62. The same was true of the “notification_open” event:



VI. GOOGLE'S BIGQUERY DATA WAREHOUSE

63. Data collected by Firebase is in part stored in the Google BigQuery data warehouse. An exemplary list of the data fields, content extracted for the March 14-16, 2023 period, and the SQL statement used to extract the data, are included in Exhibit I-7.

64. I compared the data stored in BigQuery (made available to app developers) to the BaseView log data produced by Google on February 23, 2023 (not made available to app developers), focusing on the January 25, 2023 data records.

65. Some data produced as part of the [REDACTED] log production was not present in the BigQuery data warehouse. For example,

- The Adwords and campaign related data shown in the [REDACTED] logs were not present in BigQuery.

- At least, the [REDACTED],

[REDACTED] nt, and

[REDACTED] were not present in BigQuery.

66. Relatedly, some data produced as part of the production from the [REDACTED] log on January 20 and January 31, 2023 were absent from Bigquery, including:

- [REDACTED]s such as [REDACTED],
[REDACTED]
- App-related [REDACTED] [REDACTED] [REDACTED] [REDACTED]
[REDACTED]
- [REDACTED]

67. Accordingly, with Firebase, Google collects data that it does not even share with app developers.